

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЧЕРКАСЬКИЙ ДЕРЖАВНИЙ ТЕХНОЛОГІЧНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ І СИСТЕМ

МЕТОДИЧНІ РЕКОМЕНДАЦІЇ

до самостійної роботи

з дисципліни «Архітектура інформаційних систем і технологій»

для здобувачів освітнього ступеня «бакалавр»

зі спеціальності 126 Інформаційні системи та технології

(освітньої програми «Web-технології, Web-дизайн»)

усіх форм навчання

Черкаси
2020

УДК 004(07)
М 54

*Затверджено вченою радою ФІТІС,
протокол № 5 від 17.02.2020 р.,
згідно з рішенням кафедри інформаційних
технологій проектування,
протокол № 8 від 10.01.2020 р.*

Упорядник: Єгорова О. В., к.т.н., доцент

Рецензент: Лавданський А.О., к.т.н., доцент

М54 Методичні рекомендації до самостійної роботи з дисципліни «Архітектура інформаційних систем і технологій» для здобувачів освітнього ступеня «бакалавр» зі спеціальності 126 Інформаційні системи та технології (освітньої програми «Web-технології, Web-дизайн») усіх форм навчання [Електронний ресурс] / [упоряд. Єгорова О. В.] ; М-во освіти і науки України, Черкас. держ. технол. ун-т. Черкаси: ЧДТУ, 2020. – 12 с. – Назва з титульного екрана.

Методичні рекомендації спрямовані на закріплення та поглиблене вивчення тем дисципліни, набуття навичок самостійного мислення та розробки архітектури програмної системи здобувачами освітнього ступеня «бакалавр» за спеціальністю 126 «Інформаційні системи та технології» (освітньої програми «Web-технології, Web-дизайн»).

УДК 004(07)

Виробничо-практичне
електронне видання
комбінованого використання

МЕТОДИЧНІ РЕКОМЕНДАЦІЇ до самостійної роботи

з дисципліни «Архітектура інформаційних систем і технологій»
для здобувачів освітнього ступеня «бакалавр»
зі спеціальності 126 Інформаційні системи та технології
(освітньої програми «Web-технології, Web-дизайн»)
усіх форм навчання

Упорядник: **Єгорова Ольга В'ячеславівна**

В авторській редакції.

ЗМІСТ

ВСТУП.....	4
ОРГАНІЗАЦІЯ САМОСТІЙНОЇ РОБОТИ	5
САМОСТІЙНА РОБОТА № 1	6
САМОСТІЙНА РОБОТА № 2	7
САМОСТІЙНА РОБОТА № 3	8
САМОСТІЙНА РОБОТА № 4	9
САМОСТІЙНА РОБОТА № 5	10
САМОСТІЙНА РОБОТА № 6	11
ЛІТЕРАТУРА.....	12

ВСТУП

Навчальна дисципліна «Архітектура інформаційних систем і технологій» належить до циклу дисциплін професійної підготовки здобувачів бакалаврського освітнього рівня за спеціальністю 126 «Інформаційні системи та технології» (освітня програма «Web-технології, Web-дизайн»).

Предметом вивчення навчальної дисципліни є моделі, методи та засоби проектування архітектури інформаційних систем та технологій.

Метою викладання навчальної дисципліни «Архітектура інформаційних систем і технологій» є теоретична та практична підготовка здобувачів бакалаврського освітнього рівня у напрямку вивчення принципів побудови інформаційних систем і технологій, архітектури, моделей та ресурсів інформаційних систем, елементного базису інформаційних систем і технологій.

Основними завданнями вивчення дисципліни «Архітектура інформаційних систем і технологій» є забезпечити розуміння та засвоєння здобувачами бакалаврського освітнього рівня базових аспектів використання архітектурних та деталізованих рішень при проектуванні систем; оволодіння здобувачами вищої освіти засобами розробки архітектури інформаційних систем та засобами розробки інформаційних систем.

Навчальна програма дисципліни містить такі змістовні модулі:

- архітектура інформаційних систем;
- архітектура інформаційних технологій.

Тематичне планування самостійної роботи здобувачів бакалаврського освітнього рівня із дисципліни «Архітектура інформаційних систем і технологій» охоплює такі теми:

- моделювання поведінки програмної системи засобами UML;
- перехід від монолітної архітектури програмної системи до розподіленої;
- on-premise рішення ESB;
- міжпроцесна взаємодія в мікросервісній архітектурі;
- реактивні мікросервісна архітектура;
- еволюційна архітектура.

ОРГАНІЗАЦІЯ САМОСТІЙНОЇ РОБОТИ

Самостійна робота здобувачів вищої освіти безпосередньо впливає на глибину, неформальність та стійкість набутих знань і умінь. Вона є основним способом оволодіння навчальним матеріалом у вільний від аудиторних занять час.

Мета і зміст самостійної роботи здобувачів бакалаврського освітнього рівня із дисципліни «Архітектура інформаційних систем і технологій» визначені навчальною і робочою програмами цієї дисципліни, методичними матеріалами.

Мета самостійної роботи полягає у забезпеченні розуміння і засвоєння здобувачами бакалаврського освітнього рівня навчальної програми дисципліни у повному обсязі шляхом свідомого закріплення, поглиблення й систематизації набутих теоретичних знань, а також набуття практичних навичок розробки архітектур програмних систем.

Методичні рекомендації до самостійної роботи здобувачів бакалаврського освітнього рівня із дисципліни «Архітектура інформаційних систем і технологій» містять шість самостійних робіт, які охоплюють основні теми вказаного курсу. Під час виконання самостійної роботи здобувачі вивчають матеріали окремих тем шляхом опрацювання відповідної літератури, виконують ситуативні завдання, здійснюють підготовку до лабораторних робіт та поточного контролю знань.

Контроль виконання самостійної роботи здійснюється у формі співбесіди, де розглядаються результати виконання ситуативних завдань та рівень засвоєння здобувачем теоретичних знань.

Знання здобувачів бакалаврського освітнього рівня із самостійної роботи оцінюються за наступними критеріями:

- високий понятійний рівень;
- глибина, широта, повнота питання, що висвітлюється;
- осмисленість і усвідомленість затверджених положень теми, яка розкривається здобувачем;
- логічність та послідовність викладу під час відповіді;
- використання та аналіз моделей, методів, засобів, які вивчалися;
- самостійність мислення;
- впевненість в правоті своїх суджень;
- вміння виділяти головне;
- вміння встановлювати міжпредметні та внутрішньопредметні зв'язки;
- вміння робити висновки, показувати перспективу розвитку ідеї або проблеми;
- стилістично грамотна мова.

САМОСТІЙНА РОБОТА № 1

Моделювання поведінки програмної системи засобами UML

Мета роботи: Формування знань та набуття практичних навичок моделювання поведінки програмної системи засобами UML.

Порядок виконання роботи

1. Опрацювати питання для самостійного вивчення.
2. Виконати ситуативне завдання.
3. Відповісти на питання для самоконтролю.
4. Пройти співбесіду за результатами виконання самостійної роботи.

Перелік питань для самостійного опрацювання

1. Прецеденти використання системи.
2. Діаграма прецедентів.
3. Організація прецедентів.
4. Створення прецедентів із використанням case-засобів.
5. Специфікації прецедентів.
6. Діаграми діяльності.
7. Попередній архітектурний аналіз системи.

Рекомендована література: [1],[2],[3].

Ситуативні завдання

Із використанням case-засобу розробити модель програмної системи автоматизації роботи пункту прокату автомобілів.

Питання для самоконтролю

1. Яким шляхом вводять обмеження у мові UML.
2. Для чого використовують представлення прецедентів?
3. Для чого використовують логічне представлення?
4. Для чого використовують представлення процесів?
5. Для чого використовують представлення компонентів?
6. Для чого використовують представлення розміщення?
7. Для чого використовують діаграму прецедентів?
8. Які відношення застосовують на діаграмі прецедентів?
9. Що таке моделювання контексту системи?
10. Що таке моделювання вимог до системи?
11. Що таке специфікація прецеденту?
12. Опишіть шаблон специфікації прецеденту.
13. Для чого використовують діаграму видів діяльності?
14. Дайте визначення стану дії.
15. Дайте визначення стану виду діяльності.

САМОСТІЙНА РОБОТА № 2

Перехід від монолітної архітектури програмної системи до розподіленої

Мета роботи: Формування знань та набуття практичних навичок переходу від монолітної архітектури до розподіленої.

Порядок виконання роботи

1. Опрацювати питання для самостійного вивчення.
2. Виконати ситуативне завдання.
3. Відповісти на питання для самоконтролю.
4. Пройти співбесіду за результатами виконання самостійної роботи.

Перелік питань для самостійного опрацювання

1. Теорема CAP
2. Проблеми масштабування.
3. Масштабування навантаження на читання.
4. CRUD.
5. Горизонтальне масштабування.
6. Синхронізація сховищ.
7. Перехід від реляційних СУБД до нереляційних.

Рекомендована література: [4],[5],[6].

Ситуативні завдання

Виконати перехід від розробленої монолітної архітектури програмної системи автоматизації роботи пункту прокату автомобілів до розподіленої.

Питання для самоконтролю

1. Дайте означення монолітної архітектури програмної системи.
2. Дайте означення розподіленої архітектури програмної системи.
3. Назвіть переваги та недоліки монолітної архітектури програмної системи.
4. Назвіть переваги та недоліки розподіленої архітектури програмної системи.
5. В чому полягає суть теореми CAP?
6. В чому полягає суть проблеми масштабування?
7. Опишіть процес масштабування навантаження на читання.
8. Дайте характеристику відомим способам вирішення проблеми CRUD.
9. Які існують підходи до виконання горизонтального масштабування?
10. Яким чином здійснюється синхронізація сховищ?
11. Опишіть процес переходу від реляційних СУБД до нереляційних.

САМОСТІЙНА РОБОТА № 3

On-premise рішення ESB

Мета роботи: Формування знань та набуття практичних навичок використання on-premise рішень ESB.

Порядок виконання роботи

1. Опрацювати питання для самостійного вивчення.
2. Виконати ситуативне завдання.
3. Відповісти на питання для самоконтролю.
4. Пройти співбесіду за результатами виконання самостійної роботи.

Перелік питань для самостійного опрацювання

1. RabbitMQ.
2. Kafka.
3. IBM Integration Bus (IBM).
4. Oracle Service Bus (Oracle).
5. BizTalk (Microsoft).
6. ActiveMatrix Service Bus (TIBCO).
7. MuleESB (MuleSoft).
8. JBoss Fuse ESB (Red Hat).

Рекомендована література: [7],[8],[9].

Ситуативні завдання

Імплементувати одне із on-premise рішень ESB в розподілену архітектуру програмної системи автоматизації роботи пункту прокату автомобілів.

Питання для самоконтролю

1. Дайте характеристику RabbitMQ.
2. Поясніть особливості використання Kafka.
3. Вкажіть переваги IBM Integration Bus (IBM).
4. Дайте характеристику Oracle Service Bus (Oracle).
5. Вкажіть особливості використання BizTalk (Microsoft).
6. В яких випадках доречно використовувати ActiveMatrix Service Bus (TIBCO).
7. Вкажіть переваги та недоліки MuleESB (MuleSoft).
8. Дайте характеристику JBoss Fuse ESB (Red Hat).

САМОСТІЙНА РОБОТА № 4

Міжпроцесна взаємодія в мікросервісній архітектурі

Мета роботи: Формування знань та набуття практичних навичок розробки мікросервісної архітектури програмної системи.

Порядок виконання роботи

1. Опрацювати питання для самостійного вивчення.
2. Виконати ситуативне завдання.
3. Відповісти на питання для самоконтролю.
4. Пройти співбесіду за результатами виконання самостійної роботи.

Перелік питань для самостійного опрацювання

1. Стили міжпроцесної взаємодії в мікросервісній архітектурі.
2. Опис API в мікросервісній архітектурі.
3. API, що розвиваються, в мікросервісній архітектурі.
4. Формати повідомлень в мікросервісній архітектурі.
5. Взаємодія на основі віддаленого виклику процедур.
6. Робота в умовах часткової відмови із застосуванням шаблону «попереджувач».
7. Взаємодія за допомогою асинхронного обміну повідомленнями.
8. Використання асинхронного обміну повідомленнями для покращення доступності.

Рекомендована література: [4],[5].

Ситуативні завдання

Розробити мікросервісну архітектуру програмної системи автоматизації роботи пункту прокату автомобілів до розподіленої.

Питання для самоконтролю

1. Назвіть стилі взаємодії в мікросервісній архітектурі.
2. Опишіть API в мікросервісній архітектурі.
3. Опишіть API, що розвиваються, в мікросервісній архітектурі.
4. Назвіть формати повідомлень в мікросервісній архітектурі.
5. Яким чином відбувається взаємодія на основі віддаленого виклику процедур?
6. Поясніть призначення REST та gRPC.
7. Опишіть роботу в умовах часткової відмови із застосуванням шаблону «попереджувач» у мікросервісній архітектурі.
8. Яким чином відбувається взаємодія під час асинхронного обміну повідомленнями в мікросервісній архітектурі?
9. Поясніть призначення асинхронного обміну повідомленнями в мікросервісній архітектурі.

САМОСТІЙНА РОБОТА № 5

Реактивна мікросервісна архітектура

Мета роботи: Формування знань та набуття практичних навичок розробки реактивної мікросервісної архітектури програмної системи.

Порядок виконання роботи

1. Опрацювати питання для самостійного вивчення.
2. Виконати ситуативне завдання.
3. Відповісти на питання для самоконтролю.
4. Пройти співбесіду за результатами виконання самостійної роботи.

Перелік питань для самостійного опрацювання

1. Реактивне програмування.
2. Поняття реактивної системи.
3. Реактивні мікросервіси.
4. Процес розробки реактивних мікросервісів.
5. Шина повідомлень Vert.x.
6. Модель асинхронного розвитку.
7. Будівельні блоки – вертикалі.
8. Реалізація HTTP-мікросервісів.
9. Використання мікросервісів HTTP.
10. Мікросервіси на основі повідомлень.
11. Ініціювання взаємодій на основі повідомлень.

Рекомендована література: [4],[5],[10].

Ситуативні завдання

Розробити реактивну мікросервісну архітектуру програмної системи автоматизації роботи пункту прокату автомобілів до розподіленої.

Питання для самоконтролю

1. Дайте означення поняття реактивне програмування.
2. Наведіть означення реактивної системи.
3. Поясніть призначення реактивних мікросервісів.
4. Опишіть процес розробки реактивних мікросервісів.
5. Дайте означення шини повідомлень Vert.x.
6. Опишіть модель асинхронного розвитку.
7. Опишіть будівельні блоки – вертикалі.
8. Яким чином реалізовані HTTP-мікросервіси?
9. Поясніть призначення мікросервісів HTTP.
10. Дайте характеристику мікросервісів на основі повідомлень.
11. Яким чином відбувається ініціювання взаємодій на основі повідомлень?

САМОСТІЙНА РОБОТА № 6

Еволюційна архітектура

Мета роботи: Формування знань та набуття практичних навичок розробки еволюційної архітектури програмної системи.

Порядок виконання роботи

1. Опрацювати питання для самостійного вивчення.
2. Виконати ситуативне завдання.
3. Відповісти на питання для самоконтролю.
4. Пройти співбесіду за результатами виконання самостійної роботи.

Перелік питань для самостійного опрацювання

1. Архітектура з еволюційним розвитком.
2. Проектування інкрементних змін.
3. Еволюція архітектурних стилів.
4. Побудова архітектури з еволюційним розвитком.
5. Антипатерни архітектури з еволюційним розвитком.
6. Впровадження еволюційної архітектури.

Рекомендована література: [5],[11].

Ситуативні завдання

Розробити еволюційну архітектуру програмної системи автоматизації роботи пункту прокату автомобілів до розподіленої.

Питання для самоконтролю

1. Поясніть сутність архітектури з еволюційним розвитком.
2. Яким чином проектуються інкрементні зміни?
3. Дайте означення функції придатності.
4. Дайте характеристику відомим архітектурним стилям.
5. Опишіть процес еволюційного проектування баз даних.
6. Яким чином відбувається інтеграція баз даних загального використання?
7. Дайте характеристику техніками побудови архітектури з еволюційним розвитком.
8. Опишіть процес побудови архітектури з еволюційним розвитком.
9. Дайте характеристику антипатернам архітектури з еволюційним розвитком.
10. Опишіть процес міграції архітектур.
11. Опишіть процес впровадження еволюційної архітектури.

ЛІТЕРАТУРА

1. Дудзяний І. М. Об'єктно-орієнтоване моделювання програмних систем: навчальний посібник. Львів: Видавничий центр ЛНУ імені Івана Франка, 2007. 108 с.
2. Дубовой В. М., Москвіна С. М., Никитенко О. Д. Моделювання процесів і систем керування. Вінниця: ВНТУ, 2009. 103 с.
3. Орлов С. А. Программная инженерия; 5-е изд., обновленное и доп. СПб: Издательский дом «Питер», 2016. 640 с.
4. Ричардсон К. Микросервисы. Паттерны разработки и рефакторинга. СПб.: Питер, 2019. 544 с.: ил. (Серия «Библиотека программиста»).
5. Wolff E. Microservices Flexible Software Architectures. Available at: <http://leanpub.com/microservices-book> (accessed 2019-09-01).
6. Переход от монолитной архитектуры к распределенной. URL: <https://blog.byndyu.ru/2014/05/blog-post.html> (дата обращения: 17.10.2019).
7. Обзор iPaaS платформы MuleSoft Anypoint. URL: <https://dou.ua/lenta/articles/ipaas-mulesoft/> (дата обращения: 17.10.2019).
8. Best Enterprise Service Bus (ESB) Software. URL: <https://www.g2.com/categories/enterprise-service-bus-esb> (accessed 2019-10-01).
9. Enterprise Service Bus (ESB) Tools: Technical Comparison and Review. URL: <https://shadow-soft.com/enterprise-service-bus-esb-tools/> (accessed 2019-10-11).
10. Escoffier C. Building Reactive Microservices in Java. Sebastopol : O'Reilly Media. 2017. 83 p.
11. Форд Н., Парсонс Р., Куа П. Эволюционная архитектура. Поддержка непрерывных изменений. СПб.: Питер, 2019. 272 с.